METHODS AND APPARATUS FOR INTERLEAVING IN A BLOCK-COHERENT COMMUNICATION SYSTEM

Related Applications

The present application claims the benefit of U.S. Provisional Patent Application S.N. 60/459,677, filed April 2, 2003 titled "METHODS AND APPARATUS FOR DATA TRANSMISSION IN A BLOCK-COHERENT COMMUNICATION SYSTEM".

Field Of the Invention

The present invention is directed to methods and apparatus for data communication over a block-coherent channel and, more particularly, to methods and apparatus for accessing and/or interleaving data bits coded by parity check codes, e.g., low-density parity-check (LDPC) codes.

Background

A communication system requires noncoherent detection when it is infeasible for the receiver to maintain a reliable estimate for instantaneous channel gain (magnitude and, especially, phase). Noncoherent communication systems include, for instance, a wireless multiple access system where the mobile, having limited power, cannot afford to transmit high power known symbols (pilots) to enable reliable channel estimation. A noncoherent communication channel may possess some coherence property: A coherent block is a time interval during which the channel variations are negligibly small. Communication over such a channel is referred to as *block-coherent* communication.

Block-coherent communication may arise naturally in fast frequency-hopping orthogonal frequency division multiple (OFDM) access systems. In such systems information may be modulated onto a subset of available frequencies, called tones, in every symbol time. To enhance spectral efficiency and increase diversity gain, tones utilized are, in some cases, rapidly hopped across the entire utilized frequency band in every L symbols, i.e., L consecutive symbols are mapped to one tone, followed by another L symbols mapped to a different tone, and so on.

When L is small, we can assume consecutive L symbols experience identical channel gain. Although the amplitudes of the gains of two consecutive L symbols can be close, their phases are normally completely independent.

More precisely, a block-coherent communication system can be defined as follows: for a system represented in discrete time domain, the channel gain is an unknown complex random variable that remains the same for every L consecutive symbols but otherwise varies independently according to some distribution, e.g., the phase is uniformly distributed over [0, 2PI] and the magnitude is Rayleigh distributed.

For block-coherent communication the nominal modulation scheme is differential M-array phase-shift-keying (DMPSK). DMPSK carries the information in the phase differences between two successive symbols over the coherent block. For illustration, to transmit N x (L-1) MPSK information symbols s(i), each of the N consecutive sets of L-1 symbols, denoted as s(1), s(2), …, s(L-1), is differentially encoded to transmitted symbols t(1), t(2), …, t(N), where t(1)=1, and t(j+1) = t(j) x s(j) for j=1, …, N-1.

Modulations other than DMPSK are possible. For instance, with the insertion of known symbols in a block, information symbols may be transmitted directly on other symbols instead of differentially. This may be referred to as pseudo-pilot modulation. It is apparent, however, that at most L-1 information symbols can be transmitted inside a dwell of length L due to the phase uncertainty. In general we assume there are M information symbols transmitted in each dwell, which can be regarded as a transmission unit. In other words, each transmission unit includes L-M known symbols. M is at most L-1. For instance, both schemes mentioned have M=L-1.

With forward error-correction coding a block-coherent communication system will normally include an encoder (which inserts structured redundancy into original data stream), a DMPSK modulator (which maps binary data bits to MPSK symbols), a demodulator (which extracts out soft information differentially and feeds it to the decoder), and a decoder (which decodes the original message based on soft information from the demodulator).

In most coded systems, a receiver applying iterative demodulation and decoding - a scheme henceforth referred to as *turbo equalization* - has significant performance gain over a non-iterative receiver. For instance, convolution and/or turbo coded DMPSK systems, investigated by Shamai et al. in "Iterative decoding for coded noncoherent MPSK communications over phase-noisy AWGN channel" published in IEE Proceedings Communication 2000, demonstrates turbo-equalization performance within 1.3 dB of channel capacity and 1 dB better than traditional schemes.

It has been shown that for turbo equalization to be maximally effective, the code design needs to take the effect of iterative demodulation into account. The necessity of code design and an effective way of achieving it are described in Jin and Richardson's paper "Design of Low-Density Parity-Check Codes in Noncoherent Communication," published in International symposium on information theory June 30 2002. The approach therein improves the performance to within 0.7 dB of channel capacity.

For optimal performance of turbo equalization, the coded bits also should be sufficiently interleaved in transmission. The interleaving serves two purposes. On one hand, it can effectively eliminate the correlation among soft-messages associated with coded bits of small distance in the block structure for decoding purposes. On the other hand, interleaving can eliminate the dependence among the messages feed back to the same coherent block for demodulation. A commonly conceived interleaving structure is *random* interleaving. To achieve random interleaving, however, the same (randomly generated) permutation must be accessible, e.g., stored, at both the transmitter and the receiver. This inflicts large memory requirements for practical systems coded by large block codes.

While the performance of turbo equalization is important, for a communication system to be practical for use in a wide range of devices, e.g., consumer devices, it is important that the turbo equalizations be capable of being implemented at reasonable cost. Accordingly, the ability to efficiently implement turbo equalization schemes used for a block-coherent communication system, e.g., in terms of hardware costs, can be important.

The usage of LDPC codes as a coding scheme in block-coherent communication is desirable given its near-capacity coding gain and rich design space.

LDPC codes are often represented by bipartite graphs, called Tanner graphs, in which one set of nodes, the *variable* nodes, correspond to bits of the codeword and the other set of nodes, the *constraint* nodes, sometimes called *check* nodes, correspond to the set of parity-check constraints which define the code. Edges in the graph connect variable nodes to constraint nodes. A variable node and a constraint node are said to be *neighbors* if they are connected by an edge in the graph. For simplicity, we generally assume that a pair of nodes is connected by at most one edge.

A bit sequence associated one-to-one with the variable nodes is a codeword of the code if and only if, for each constraint node, the bits neighboring the constraint (via their association with variable nodes) sum to zero modulo two, i.e., they comprise an even number of ones.

The number of edges attached to a node, i.e., a variable node or a constraint node, is referred to as the *degree* of the node. A *regular* graph or code is one for which all variable nodes have the same degree, j say, and all constraint nodes have the same degree, k say. In this case we say that the code is a (j,k) regular code. These codes were originally invented by Gallager (1961). In contrast to a "regular" code, an irregular code has constraint nodes and/or variable nodes of differing degrees. For example, some variable nodes may be of degree 4, others of degree 3 and still others of degree 2.

While irregular codes can be more complicated to represent and/or implement, it has been shown that irregular LDPC codes can provide superior error correction/detection performance when compared to regular LDPC codes.

An exemplary bipartite graph 200 determining a (3,6) regular LDPC code of length ten and rate one-half is shown in Fig. 2. Length ten indicates that there are ten variable nodes V_1-V_{10}, each identified with one bit of the codeword X_1-X_{10}, the set of variable nodes V_1-V_{10} is generally identified in Fig. 2 by reference numeral 202. Rate one half indicates that there are half as many check nodes as variable nodes, i.e., there are five check nodes C_1-C_5 identified by reference numeral 206. Rate one half further indicates that the five constraints are linearly independent, as discussed below.

While Fig. 2 illustrates the graph associated with a code of length 10, it can be appreciated that representing the graph for a codeword of length 1000 would be 100 times more complicated.

The practical challenges posed by turbo equalization, in the light of implementation cost, are the complexity of soft-in soft-out (SISO) demodulator and the data interleaving at the transmitter and the receiver. There remains a need for improved interleaving techniques and/or implementation. The present invention is directed to implementing an interleave in an efficient manner. The data interleaving techniques of the present invention are well suited for use with data bits encoded, e.g., by Low Density Parity Check (LDPC) codes.

Brief Description of the Figures

Figure 1 illustrates an exemplary communication system including an encoder, an interleaver, a modulator, a channel, a demodulator, a deinterleaver, and a decoder in which the methods and apparatus of the present invention are employed.

Figure 2 illustrates a bipartite graph representation of an exemplary regular LDPC code of length ten.

Figure 3 is a graphical representation of a small LDPC code that is used as the basis of a much larger LDPC code to present an example in accordance with the present invention.

Figure 4 graphically illustrates the effect of making three copies of the small LDPC graph shown in Fig.3.

Figure 5 illustrates the result of algebraic interleaving by using an exemplary 4 x 8 LDPC code in accordance with the present invention. In particular, it illustrates the direct mapping of coded data bits to a portion of a transmission unit.

Figure 6 illustrates an exemplary interleaver, suitable for use as the interleaver at the transmitter of the system shown in Fig. 1, which implements algebraic interleaving in accordance with the present invention.

Figure 7 illustrates the correspondence between soft values from the LDPC decoder and the transmission units using the exemplary code structure shown in Fig.5. The soft values include 3-bits.

Figure 8 illustrates an exemplary structure for applying interleaving and deinterleaving in iterative demodulation and decoding, suitable for use as the interleaver and the deinterleaver at the receiver of the system shown in Fig. 1.

Summary of the Invention

The present invention is directed to methods and apparatus for data communication over a block-coherent channel. For simplicity, we refer to the symbols transmitted in one coherent interval as a *dwell*. The length of a dwell is L. The symbols of a dwell, serve as an information transmission unit, that includes M information symbols and (L-M) known symbols. Each information symbol is a symbol mapped from P-bit(s) in a codeword.

The present invention is directed to methods and apparatus for algebraically interleaving coded bits between the encoder and modulator at the transmitter with the use of no or relatively little memory. The present invention is also directed to methods and apparatus for algebraically interleaving soft messages between a decoder and demodulator at the receiver with no or relatively little memory.

For purposes of explaining the present invention, we assume QPSK as the modulation constellation. With this assumption, one transmission symbol uses two coded bits – a dibit. The techniques described in the present invention, however, can be easily generalized and are applicable to high order modulation schemes, e.g. MPSK or QAM with such implementations being deemed within the scope of the present invention.

The present invention of algebraic interleaving is directed to data bits encoded by LDPC codes that possess a certain hierarchical structure in which a full LDPC graph appears to be, in large part, made up of multiple copies, Z say, of a Z times smaller graph.

The Z graph copies may be identical. To be precise we will refer to the smaller graph as the *projected* graph. We refer to the Z parallel edges as vector edges, and Z parallel nodes as vector nodes. In a previous application, U.S. Patent Application S.N.09/975,331 titled "Methods and Apparatus for Performing LDPC Code Encoding and Decoding", filed October 10, 2001, which is hereby expressly incorporated by reference, the benefits that such a structure lends to decoder implementation were described. A key observation is that all operations may be done in parallel across all copies. The Z copies are not disjoint, however, they are combined to form one large graph, Z times larger than the projected graph. This is accomplished by interconnecting the Z copies of the projected graph in a controlled manner. Specifically, we allow the Z edges within a vector edge to undergo a permutation, or exchange, between copies of the projected graph as they go, e.g., from the variable node side to the constraint node side. In the vectorized encoding process corresponding to the Z parallel projected graphs, we allow the Z bits within a vector corresponding to a projected variable node to undergo a permutation.

The objective of interleaving between coding and modulation is to mitigate correlation between soft messages (values) corresponding to coded bits of short distance inside the LDPC graph. In addition, it may be equally desirable that the interleaving may create structures that assist turbo equalization. For this purpose, the coded bits inside a dwell are preferred to have a wide range of degrees, because it is known that the convergence speed of nodes of different degrees vary significantly in the decoding process. When decoded successfully in earlier phases, bits associated with high degrees can generate more reliable estimation on the unknown phase inside a dwell. This generally improves the soft messages generated on its neighboring symbols inside the dwell, which in turn assists the decoder more effectively.

This desirable property of a dwell having a wide-range mixture of node degrees can be readily implemented when data bits are encoded by vector-LDPC codes, which possess an exploitable structure. If the memory storing those encoded bits is configured as a matrix of Z

x n, then the bits in the same column (physically the same address) correspond to the same node in the projected graph and thus have the same degree.

The vectorized encoding process, as described in the U.S Patent Application S.N. _____ titled "METHODS AND APPARATUS FOR ENCODING LDPC CODES" filed July 11, 2003, may arrange these columns such that the degrees of their associated variable nodes are in an increasing order. Such an ordering facilitates an algebraic interleaving method that forms dwells with coded bits associated with variable nodes of wide range of degrees. An exemplary way is to uniformly divide the matrix into L-1 contiguous sub-matrices. The property that columns are ordered in increasing degree ensures that the sub-matrices contain variable nodes of similar degrees: the first sub matrix has the lowest degree, the last sub matrix the highest. Thus a dwell formed by taking two bits from different sub-matrices has the desired property.

Interleaving of the coded bits, implemented in accordance with the present invention may be performed as discussed below.

An exemplary interleaver apparatus of the invention includes a memory for storing coded bits and an interleaving circuit. The interleaving circuit generates a set of control information which is used to control the reading of bits from the memory. Each set of generated control information includes a transmission unit identifier, a Z vector identifier, and a row identifier. The control circuit, in an exemplary embodiment, includes four components: a symbol ID generation module, a bits ID generation module, a column ID generation module and a control information generator module. The symbol ID generation module may be implemented as a repeating counter that generates a number *s* ranging from 0 to M-1. The repeating counter is incremented periodically as a function of a system clock *clk*, e.g., s is incremented once per clock cycle; the number *s* determines the symbol index in a dwell. The bits ID generation module may also be implemented as a repeating counter, e.g., a repeating counter generating a number *b* ranging from 0 to z-1. The number *b* is periodically incremented each time the symbol index signal *s* reaches zero; number *b* determines the bits index selected in the column, e.g., a row of the a column in an array which is stored in memory . Column ID generation module 603 generates a number *c* ranging from 0 to a-1 and may be implemented as another counter. The number *c* is incremented each time the bits index *b* reaches zero; number *c*

is the column index. Taking bits index *b*, symbol index *s*, and column index *c*, a control information generator module produces a set of control information including a transmission unit identifier, a Z vector identifier and a row identifier used to control which location in the coded bits memory is accessed. In one exemplary embodiment, the Z-vector identifier is c + a x s, and the row identifier is 2 x b. The transmission unit is identified by a transmission unit identifier having the value b+cxZ, where b and c are as defined above and where Z is the number of elements in each Z vector and where x indicates a multiplication operation.

At the receiver side, the direct mapping between data bits and transmission units is conformed for demodulation. It is assumed that soft-outputs from the decoder have the same ordering as the binary code word structure. The soft values includes, however, K-bits corresponding to a coded bit. Each of k bits may be stored in a different one of D arrays in memory where D is a positive integer. In most cases, k is an integer multiple of D. One exemplary memory has three-bit soft values for each bit, each identified by the same code bit identifier. Those three bits might be in one memory location; or those three bits could be in three different memory location. With this structure, it is clear that the same type of interleaving apparatus used in a transmitter can also be used in a receiver to access soft values corresponding to a transmission unit for purposes of demodulation.

Assuming the received values also conform to the same ordering as a code word, then again the same interleaving structure can be applied in the process of accessing received values for demodulation.

Numerous additional embodiments, features and benefits of the methods and apparatus of the invention are discussed in the detailed description which follows.

Detailed description of the invention

Fig. 1 illustrates a general-purpose communication system 10 which implements the present invention. The system 10 includes a transmitter 100 which is coupled by a communication channel 110 to a receiver 120. The transmitter 100 comprises a data encoding circuit, e.g., an encoder 101, an interleaver 102, and a modulator 103. The receiver comprises a demodulator 121, a deinterleaver 122, an interleaver 123, and a data decoder 124. The encoder

101 maps an input binary data stream **A** to a structured binary data stream $\mathbf{X}_1$ with redundancy. The interleaver 102 interleaves $\mathbf{X}_1$ to another data stream $\mathbf{X}_2$. The modulator 103 transforms the binary stream $\mathbf{X}_2$ to physical signals feasible for practical transmission, e.g. QPSK signals. The communications channel 110 may be, e.g., an air link. Modulated signals are transmitted through the channel 110 to the receiver 120. At the receiver side, demodulator 121 extracts information $\mathbf{X}_2$' from the noisy distorted reception **Y**. The deinterleaver 122 reorders the soft messages $\mathbf{X}_2$' to $\mathbf{X}_1$' corresponding to the original ordering of the code structure. And the decoder 123 tries to recover the original binary data stream **A** through the use of redundancy present in the coded data stream $\mathbf{X}_1$' produced by demodulation. The data path from decoder 123 to demodulation 121 represents a feedback loop. The feedback messages should be interleaved by an interleaver 124 to transform the ordering at the code to the ordering at the modulation.

Fig. 3 illustrates a simple irregular LDPC code in the form of a graph 300. The code is of length five as indicated by the 5 variable nodes V_1 through V_5 302. Four check nodes C_1 through C_4 306 are coupled to the variable nodes 302 by a total of 12 edges 304.

Fig. 4 is a graph 400 illustrating the result of making 3 parallel copies of the small graph illustrated in Fig. 3. Variable nodes 402', 402" and 402"' correspond to the first through third graphs, respectively, resulting from making three copies of the Fig. 3 graph. In addition, check nodes 406', 406" and 406"' correspond to the first through third graphs, respectively, resulting from making the three copies. Note that there are no edges connecting nodes of one of the three graphs to nodes of another one of the three graphs. Accordingly, this copying process, which "lifts" the basic graph by a factor of 3, results in three disjoint identical graphs. (Normally the 3 copies are interconnected by permuting vector edges.)

We will now proceed by describing the algebraic interleaving of the coded bits, implemented in accordance with the present invention.

Coded bits are stored in memory configured as Z x n. Or equivalently, we view the binary codeword as n Z-vectors, each vector including Z bits. The *Z* used in vector LDPC codes is a multiple of P, the number of bits associated with a transmitted symbol. For assumed QPSK modulation where P=2, we have $Z=2z$. We further select the number of columns to be a

multiple of M, the number of information transmission symbols in a transmission unit, i.e. $n=aM$.. An interleaving method of the invention determines the location of the P-bits associated with each symbol in each transmission unit. Clearly, the memory location corresponding to a Z-vector identifier and the offset value inside the Z-vector. The present invention orders coded data as follows: The jth dwell (where j is from *1* to *z* x *a*) will contain 2-bits *in the Z-bit vector identified by i x n/M + [j/(Z/P)] with offset value 2* (j mod Z/P)*, where i is from *0* to *M-1*. In such a case, the address used to retrieve the data bits can be easily generated algebraically, without the use of memory for this purpose.

Fig. 5 illustrates an exemplary interleaving structure for a coherent interval where L=5, M=4, and P=2. Array 500 represents coded bits that have been encoded by a vector-LDPC having Z=4 and n=8. The array may be stored in a corresponding structured array of memory locations. The coded bits 500 are stored with a 4 x 8 configuration where each element in the array 500 represents one bit; the bit at position (i,j) is denoted by c(i,j). Each dwell 501, 502, 503 and 504 will include 4 information symbols formed by using the bits of array 500. The proposed technique makes the first dwell 801 containing coded bits c(1,1), c(2,1), c(1,3), c(2,3), c(1,5), c(2,5), c(1,7), c(2,7). For ease of illustration, the exemplary modulation used is pseudo-pilot modulation so a dibit 00 is inserted into each dwell (at the middle) providing one of the 5 symbols in the dwell. Therefore, the first transmitted dwell is S(c(1,1)c(2,1)), S(c(1,3)c(2,3)), S(00), S(c(1,5)c(2,5)), S(c(1,7)c(2,7)) as shown in 501. Similarly, the second dwell 502 comprises S(c(3,1)c(4,1)), S(c(3,3)c(4,3)), S(00), S(c(3,5)c(4,5)), S(c(3,7)c(4,7)). And so on.

An apparatus, e.g., interleaver 102 of the invention, for the proposed interleaving technique is shown in Fig. 6. The interleaver 102 includes a memory 610 for storing coded bits and an interleaving circuit 600 coupled together as shown in Fig. 6. Interleaving circuit 600 generates a set of control information which is used to control the reading of dibits from the memory 610. Each set of generated control information includes a transmission unit identifier, a Z vector identifier, and a row identifier. Circuit 600 includes four components: a symbol ID generation module 601, a bits ID generation module 602, a column ID generation module 603 and a control information generator module 604. Module 601 is a repeating counter that generates a number *s* ranging from 0 to M-1. The counter 601 is incremented periodically as a function of a system clock *clk*, e.g., s is incremented once per clock cycle; number *s* determines the symbol index in a dwell. Module 602 is a repeating counter generating a number *b*-ranging

from 0 to z-1. The number *b* is periodically incremented each time the symbol index signal *s* reaches zero; number *b* determines the bits index selected in the column, e.g., a row of the a column in array 600 which may be stored in memory 605. Module 603 is another counter. Module 603 generates a number *c* ranging from 0 to a-1. The number *c* is incremented each time the bits index *b* reaches zero; number *c* is the column index. Taking bits index *b*, symbol index *s*, and column index *c,* a control information generator module 604 produces a set of control information including a transmission unit identifier, a Z vector identifier, and a row identifier used to control which location in the coded bits memory 605 is accessed. The Z-vector identifier is c + a x s, and the row identifier is 2 x b. The transmission unit is identified by a transmission unit identifier having the value b+cxZ, where b and c are as defined above and where Z is the number of elements in each Z vector and where x represent a multiplication operation.

At the receiver side, the direct mapping between data bits and transmission units is conformed for demodulation. We assume that soft-outputs from the decoder have the same ordering as the binary code word structure, e.g., as shown in array 600. The soft values includes, however, K-bits corresponding to a coded bit. Each of k bits may be stored in a different one of D arrays where D is a positive integer. In most cases, k is an integer multiple of D. An exemplary memory 700 has three-bit soft values for each bit, each identified by the same code bit identifier. Those three bits might be in one memory location; or those three bits are in three different memory location 701, 702, 703, as shown in 700. With this structure, it is clear that the same interleaving circuit 600 can be used to access soft values for a transmission unit for demodulation.

Assuming the received values also conform to the same ordering as a code word, then again the same interleaving structure can be applied in the process of accessing received values for demodulation.

Inside turbo equalization, we assume that the soft-outputs from the vector-decoder (module 800) and soft-inputs from the channel receiver (module 808) have the same configuration as the coded bits, e.g., as shown in array 600. Entries in the above-mentioned configuration, however, will be K-bit integers instead of bits since they are soft messages. The interleaving circuit 802 may be the same as the circuit 600 shown in Fig. 6. The

interleaving circuit 802 generates the correct address to access both soft-outputs from the decoder and soft-inputs from the channel in the ordering of dwells. The same generated addresses, subject to a delay introduced by a delay line 810 , provide write addresses used to control the writing of the soft inputs to the decoder (module 806) after demodulation.

Many of the above described methods or method steps can be implemented using machine executable instructions, such as software, included in a machine readable medium such as a memory device, e.g., RAM, floppy disk, etc. to control a machine, e.g., general purpose computer with or without additional hardware, to implement all or portions of the above described methods, e.g., in one or more communications network nodes. Accordingly, among other things, the present invention is directed to machine-readable medium including machine executable instructions for causing a machine, e.g., processor and associated hardware, to perform one or more of the steps of the above-described method(s).

Numerous additional variations on the methods and apparatus of the present invention described above will be apparent to those skilled in the art in view of the above description of the invention. Such variations are to be considered within the scope of the invention. The methods and apparatus of the present invention may be, and in various embodiments are, used with CDMA, orthogonal frequency division multiplexing (OFDM), and/or various other types of communications techniques which may be used to provide wireless communications links between access nodes and mobile nodes. In some embodiments the access nodes are implemented as base stations which establish communications links with mobile nodes using OFDM and/or CDMA. In various embodiments the mobile nodes are implemented as notebook computers, personal data assistants (PDAs), or other portable devices including receiver/transmitter circuits and logic and/or routines, for implementing the methods of the present invention.